Evaluation of Machine Learning Models with Functional Selection of Climate-Based Harvest Recommendation Systems

Aniket R Rajput[1] Mrs. A. R. Garkhedkar[2]
[1,2]Department of Computer Applications
[1,2]P.E.S. Modern College of Engineering, Pune, Maharashtra, India

Abstract — The implementation of machine learning techniques to optimize crop selection based on essential soil and climate parameters. The soil attributes analyzed include pH values, nitrogen (S), phosphorus (P), and potassium (K) concentrations, while climate variables include temperature (°C), moisture (%) and precipitation (mm). By using the dataset for harvest recommendations from Kaggle, the study identifies key determinants of plant suitability and evaluates the effectiveness of several machine learning algorithms, such as accidental forests, gradient increase, vector machine (SVM) support, and KNearest Neighbor (KNN). The aim is to improve agricultural productivity through data-controlled recommendations tailored to specific environmental conditions. This study uses advanced feature selection methods to identify the primary variables that have an impact on harvest recommendations and provide valuable insight into the key factors needed for accurate predictions. Various models of the evaluation of machine learning are done using power metrics such as accuracy, recall, and F1 scores to determine the most effective approach. The findings indicate that all models compared to other algorithms exhibit strong prediction skills, with gradients and random forests consistently increasing higher accuracy and reliability. This result provides a valuable perspective for the continued development of precision agriculture and its practical implementation, which promotes a data-controlled approach to optimizing crop selection.

Keywords: Crop Selection Optimization, Precision Agriculture, Machine Learning in Agriculture, Soil and Climate Analysis, Feature Selection Techniques, Support Vector Machine (SVM), K-Nearest Neighbors (KNN)

I. INTRODUCTION

The agricultural sector plays a vital role in India, forms the basis of its economy and serves as the main source of life in rural communities. Almost 70% of rural households rely on agriculture, and 82% of farmers, classified as small and marginal, remain the largest employment providers. Furthermore, agriculture is extremely important for ensuring nutritional safety and promoting socioeconomic progress. However, the contribution to the country's total value (GVA) is gradually decreasing, highlighting the need for innovative strategies to maintain and improve productivity.

The Indian agricultural sector in India is increasingly facing challenges due to changes in climatic conditions, including irregular rain, rising temperatures and unpredictable weather patterns. These factors have led to the traditional methods being less reliable for harvesting and farming choices. Agricultural experts are quite knowledgeable, but often lack comprehensive insights into a variety of plants. This may be suitable for specific soil and climatic conditions. As a result, farmers encounter difficulties in determining a well-discovered harvest, leading to a decline in yield and economic set-off.

In order to tackle this problem, this research study uses machine learning techniques to recommend the best plants for arable land based on important soil and weather parameters. Important factors such as soil pH value, nitrogen (S), phosphorus (P), potassium (K), temperature, humidity, and precipitation are analyzed to determine the optimal crop recommendation. By using advanced predictive models such as Random Forest, Gradient Hosting, Support Vector Machine (SVM), and K-Nearest Neighbor (K-NN), this study evaluates the effectiveness of harvest selection when it occurs. By providing controlled knowledge and minimizing reliance in contrast to conventional methods, this approach can improve agricultural efficiency as well as shifts in climate resilience. Ultimately, the goal of this research is to support government goals, increase farmers' incomes, and promote sustainable agricultural practices.

II. TERMINOLOGY

1) Pandas: Pandas is a Python library developed for the efficient use, analysis and manipulation of structured data. This is particularly beneficial when dealing with tabular data formats such as CSV and Excel files. The library provides powerful indexing, data disputes, and cleaning features and is a critical tool for preparing data records with machine learning.
2) Numpy: numpy (numerical Python) is the basic library of numerical calculations in Python. It enables the creation and manipulation of large-scale multidimensional arrays, providing a wide range of mathematical functions, including statistical and algebraic operations. Numpy is vital to preparing data for machine learning applications.
3) Matplotlib: Matplotlib is an extensive Python library used for generating static, interactive, animated visualizations. It supports the creation of various types of diagrams and diagrams and is an important tool for data research and presentation.
4) Seaborn: Seaborn is built on Matplotlib and is a Python Visualization Library specializing in the production of attractive and insightful statistical graphics. With integrated topics and features, it simplifies visualization of complex data including thermal and PAARD diagrams.
5) ScikitLearn: ScikitLearn is a Python library that provides a large collection of tools for machine learning and data analysis. It provides modules for tasks such as model training, validation, and evaluation, as well as service programs for character selection and preprocessing.
6) Random Forest Classifier: This is a supervised algorithm applied to classification tasks problems. Building several decisions, building trees, and combining costs to

improve predictability and model stability. Random Forest is data-free and effective in reducing over-fit.

7) Gradient Boost Classifier: Gradient Boost is a technique in ensemble learning that boosts prediction accuracy by determining weak models, usually decisions and refined errors. It is known for its efficiency and high performance in classification tasks.
8) Support Vector Classifier (SVC): SVC is an algorithm used in machine learning that classifies data points using hyperlevels. It is especially useful for data records with many features, and various kernel functions can be employed to consider various data distributions.
9) knearest Neighbors (KNN): KNN is a straightforward classification method that assigns data points to categories relying on the majority class of the nearest neighbor. It is easy to understand and serves as a basic model for classification problems.
10) ZugTest Split: A technology that evaluates the effectiveness of machine learning models by splitting Data Record into two sub-quantities. One is for training and the other is for testing. The models are trained in training subgroups and evaluated on test data.
11) Crossvalidation: A reliable method of evaluation of machine learning models by splitting data records into several sub-quantities. Each subgroup is used for validation at least once, while the remaining data is used for training, ensuring more accurate performance measurements.
12) Classification Report: A summary that presents key metrics for classification performance, such as accuracy, recall, F1 score, and accuracy for each class in the dataset.
13) Confusion Matrix: Tables analyzing the performance of classification models are analyzed by displaying the values of real positive,+true&negative,_false#positive, false negative relatives.

III. Literature Summary

Machine learning has evolved into a critical tool for addressing agricultural challenges, particularly in harvest recommendation systems. Research has shown the effectiveness of algorithms for machine learning in the analysis of environmental and socioeconomic factors to optimize plant selection. As an illustration, earlier research analysed how climate change affects agricultural productivity and highlighted the need for adaptive strategies to integrate climate data into predictive models of plant selection. In this study, Na orthogonal Bayes, support vector machines, and K-means clustering were implemented to determine appropriate plants based on environmental factors such as temperature, bed pH, and moisture levels. The inclusion of IoT further improves the validity of these recommendations. Similarly, satellite remote exploration was used to estimate crop yields at the granular level, indicating the potential for satellite data in agricultural decisions. region.

In spite of these developments, more advanced for machine learning, such as Random Forest and xgboost, for harvest recommendations in diverse and resource-limited environments, must be explored. The aim of this research is to fill this gap by assessing and comparing these progressive algorithms using climate, soil and environmental parameters and data records. The research strives to enhance the efficiency of the harvest recommendation system by combining machine learning techniques with localized data and ultimately improving agricultural decision-making and sustainability. algorithm. The methodology used is detailed as data preprocessing, feature selection, model training, evaluation, and the following:

A. Data collection and preprocessing

1) Data Record: The data record "crop_recommendation.csv" includes features such as bed nutrients (nitrogen, phosphorus, potassium), climate parameters (temperature, moisture, precipitation), pH value, and corresponding harvest specifications.

```python
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix

# Load Dataset
df = pd.read_csv("Crop_recommendation.csv")

print(df.head())
```

```
     N   P   K  temperature   humidity        ph   rainfall       label
0  119  25  51    26.473302  80.922544  6.283818  53.657426  watermelon
1  119  19  55    25.187800  83.446217  6.818261  46.874209  watermelon
2  105  30  50    25.299547  81.775276  6.376201  57.041471  watermelon
3  114   8  50    24.746313  88.308663  6.581588  57.958261  watermelon
4   93  22  52    26.587407  81.325632  6.932740  41.875400  watermelon
```



```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   N            2200 non-null   int64  
 1   P            2200 non-null   int64  
 2   K            2200 non-null   int64  
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object 
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB
None
```

```
print(df.describe())
```

```
                N            P            K  temperature     humidity  \
count  2200.000000  2200.000000  2200.000000  2200.000000  2200.000000   
mean     50.551818    53.362727    48.149091    25.616244    71.481779   
std      36.917334    32.985883    50.647931     5.063749    22.263812   
min       0.000000     5.000000     5.000000     8.825675    14.258040   
25%      21.000000    28.000000    20.000000    22.769375    60.261953   
50%      37.000000    51.000000    32.000000    25.598693    80.473146   
75%      84.250000    68.000000    49.000000    28.561654    89.948771   
max     140.000000   145.000000   205.000000    43.675493    99.981876   

                ph     rainfall  
count  2200.000000  2200.000000  
mean      6.469480   103.463655  
std       0.773938    54.958389  
min       3.504752    20.211267  
25%       5.971693    64.551686  
50%       6.425045    94.867624  
75%       6.923643   124.267508  
max       9.935091   298.560117  
```

2) Data Cleaning: Missing values were identified, and their absence confirmed using the. isnull() and .info() methods. Descriptive statistics were reviewed to ensure data consistency.

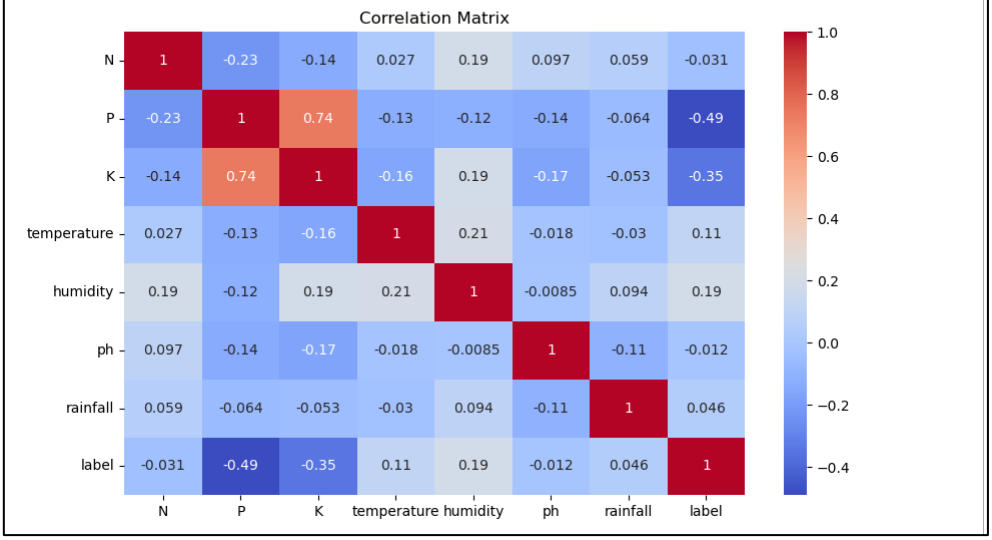
```
#Handle Missing Values
df.fillna(df.mean(), inplace=True)
```

3) Label Coding: Representative target variables (labels) representing plants were coded in numerical categories using label codes using machine learning models using label codes.

4) Exploration Data Analysis (EDA): Correlated heating caps have been applied to eliminate properties and redundancy. For detailed visualization, a PAARD diagram was optionally generated.

```
# Encode Target Variable (Label Encoding for Classification)
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df['label'] = label_encoder.fit_transform(df['label'])

# Correlation Matrix (EDA Step)
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```



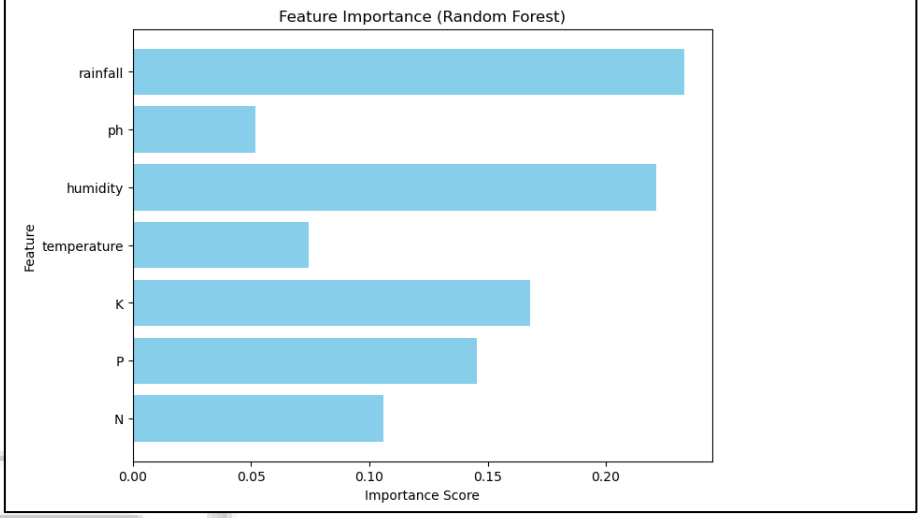
1) Feature Selection

a) Relevant features (Nitrogen, Phosphorus, Potassium, temperature, humidity, pH, and rainfall) were selected for predictive modeling.

b) The significance of features was determined using the Random Forest algorithm to prioritize influential variables, ensuring model interpretability.

```
# Feature Importance (Random Forest)
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
feature_importances = rf_model.feature_importances_

plt.figure(figsize=(8, 6))
plt.barh(X.columns, feature_importances, color='skyblue')
plt.title("Feature Importance (Random Forest)")
plt.xlabel("Importance Score")
plt.ylabel("Feature")
plt.show()
```



2) Data Splitting

The dataset was split into training (70%) and testing (30%) subsets using the train_test_split method. Stratified sampling ensured proportional distribution of crop categories.

```
# Splitting Dataset
X = df[['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall']]
y = df['label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

```

3) Model Implementation

Machine learning for four algorithms has been implemented to evaluate prediction accuracy.

a) Random Forest Classifier: Some Decisions - Create some decisions with controlled overadaptation by t he ensemble method.

b) Graduate Boost Classifier: Applied to optimize pre dictions by minimizing losses due to iterative incre ases.

c) Support Vector Machine (SVM): Consisting of an RBF kernel for nonlinear classification.

d) kNN: Used to classify plants using metrics.

4) Model evaluation

a) The model has been evaluated in tests. Data entries were assessed with metrics such as accuracy, accuracy, recall, F1 score (via classification report), and visualization of the confusion matrix.

b) Cross-validation was performed to ensure reliability of the results. For comparison, the mean cross-validation accuracy was calculated.

c) The results were visualized with a comparison bar diagram to identify the best model.


```python
# Evaluation Function
def evaluate_model(model, X_train, X_test, y_train, y_test, model_name):
    # Train the model
    model.fit(X_train, y_train)

    # Predict
    y_pred = model.predict(X_test)

    # Accuracy Score
    accuracy = accuracy_score(y_test, y_pred)
    print(f"\n{model_name} Accuracy: {accuracy * 100:.2f}%")

    # Classification Report
    print(f"\nClassification Report for {model_name}:")
    print(classification_report(y_test, y_pred))

    # Confusion Matrix
    cm = confusion_matrix(y_test, y_pred)
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
    plt.title(f"{model_name} Confusion Matrix")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.show()

    # Cross-Validation
    cv_scores = cross_val_score(model, X_train, y_train, cv=5)
    print(f"\n{model_name} Cross-Validation Accuracy: {np.mean(cv_scores) * 100:.2f}%")

    return accuracy

```

```python
# Model Training and Evaluation
models = {
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42),
    "Gradient Boosting": GradientBoostingClassifier(random_state=42),
    "SVM": SVC(kernel='rbf', C=1, random_state=42),
    "k-NN": KNeighborsClassifier(n_neighbors=5)
}

results = {}
for model_name, model in models.items():
    print(f"\n--- Evaluating {model_name} ---")
    results[model_name] = evaluate_model(model, X_train, X_test, y_train, y_test, model_name)

```

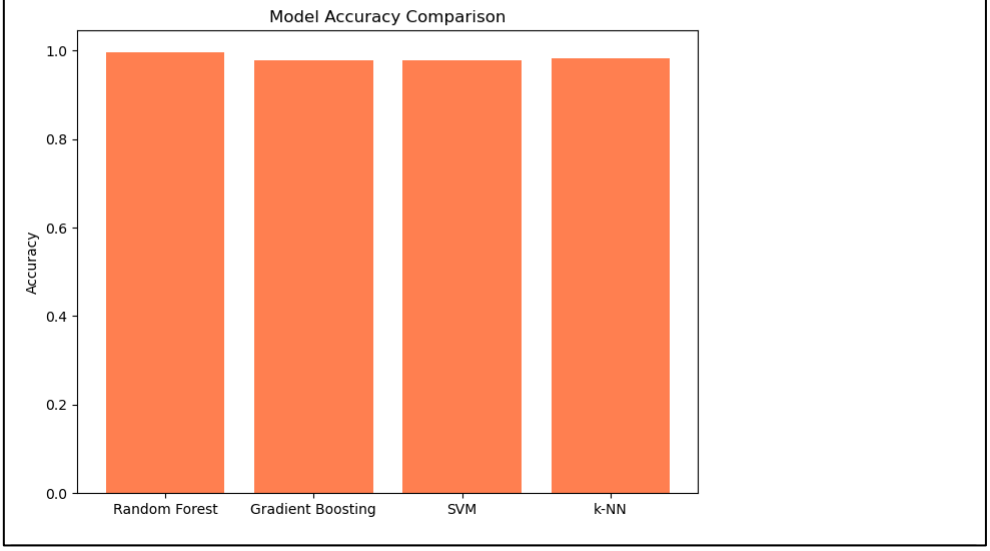
5) Model Comparison and Insights

a) A comprehensive comparison of accuracy scores across all models was performed.

b) Random Forest was further analyzed for feature importance, generating insights into the key determinants influencing crop recommendations.

```python
# Comparing Models
plt.figure(figsize=(8, 6))
plt.bar(results.keys(), results.values(), color='coral')
plt.title("Model Accuracy Comparison")
plt.ylabel("Accuracy")
plt.show()

```



IV. RESULTS:

The effectiveness of various algorithms in machine learning models, incorporating random forests, gradient hosting, SVMs, and knearest Neighbor (Knn). The model was trained using 70% of the data records, with the remaining 30% being reserved for testing. The results are represented by accuracy, confusion matrix, classification reports, and cross-validation values, providing a comprehensive assessment of the predictive performance of each model. Visualizations were

created to analyze the properties and compare the output of the model.

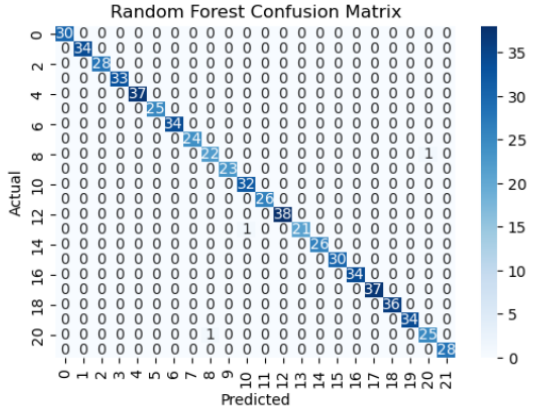
A. Model Performance

1) Random Forest Classifier` Accuracy:

The Random Forest model achieved an accuracy of 98.95%. Precipitation, pH values and temperature were provided as the most significant factors influencing harvest recommendations.

Classification Report for Random Forest:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	30
1	1.00	1.00	1.00	34
2	1.00	1.00	1.00	28
3	1.00	1.00	1.00	33
4	1.00	1.00	1.00	37
5	1.00	1.00	1.00	25
6	1.00	1.00	1.00	34
7	1.00	1.00	1.00	24
8	0.96	0.96	0.96	23
9	1.00	1.00	1.00	23
10	0.97	1.00	0.98	32
11	1.00	1.00	1.00	26
12	1.00	1.00	1.00	38
13	1.00	0.95	0.98	22
14	1.00	1.00	1.00	26
15	1.00	1.00	1.00	30
16	1.00	1.00	1.00	34
17	1.00	1.00	1.00	37
18	1.00	1.00	1.00	36
19	1.00	1.00	1.00	34
20	0.96	0.96	0.96	26
21	1.00	1.00	1.00	28
accuracy			1.00	660
macro avg	0.99	0.99	0.99	660
weighted avg	1.00	1.00	1.00	660



Random Forest Cross-Validation Accuracy: 99.29%

B. Gradient boost classifier:

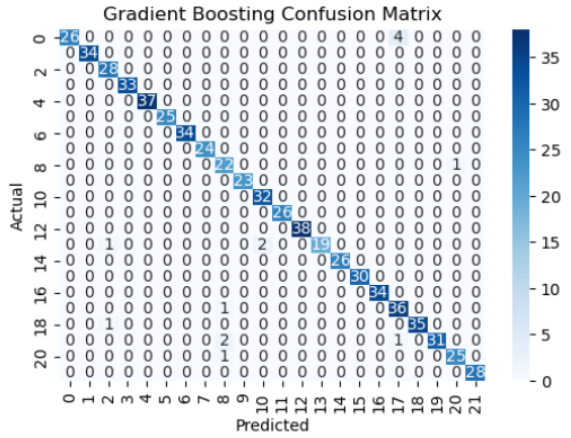
The increase in graduates achieved an accuracy of 97.45%, which indicates strong prediction skills. It is identified as the most influential characteristic.

--- Evaluating Gradient Boosting ---

Gradient Boosting Accuracy: 97.88%

Classification Report for Gradient Boosting:

	precision	recall	f1-score	support
0	1.00	0.87	0.93	30
1	1.00	1.00	1.00	34
2	0.93	1.00	0.97	28
3	1.00	1.00	1.00	33
4	1.00	1.00	1.00	37
5	1.00	1.00	1.00	25
6	1.00	1.00	1.00	34
7	1.00	1.00	1.00	24
8	0.85	0.96	0.90	23
9	1.00	1.00	1.00	23
10	0.94	1.00	0.97	32
11	1.00	1.00	1.00	26
12	1.00	1.00	1.00	38
13	1.00	0.86	0.93	22
14	1.00	1.00	1.00	26
15	1.00	1.00	1.00	30
16	1.00	1.00	1.00	34
17	0.88	0.97	0.92	37
18	1.00	0.97	0.99	36
19	1.00	0.91	0.95	34
20	0.96	0.96	0.96	26
21	1.00	1.00	1.00	28
accuracy			0.98	660
macro avg	0.98	0.98	0.98	660
weighted avg	0.98	0.98	0.98	660



Gradient Boosting Cross-Validation Accuracy: 98.25%

1) Support Vector Machine (SVM):

- Accuracy: SVM produced an accuracy of 92.83%, indicating good performance but less reliable compared to ensemble methods.
- Cross-Validation Accuracy: The cross-validation accuracy was 91.50%, reflecting slight variability in performance across training subsets.
- Confusion Matrix: SVM exhibited more misclassifications for certain crop categories, suggesting a potential limitation in handling complex relationships.

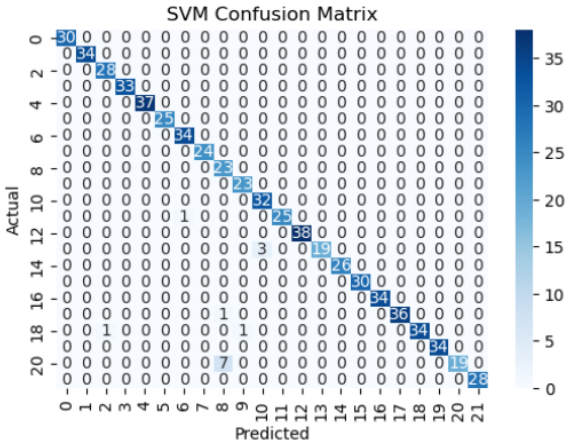

```
--- Evaluating SVM ---

SVM Accuracy: 97.88%

Classification Report for SVM:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        30
           1       1.00      1.00      1.00        34
           2       0.97      1.00      0.98        28
           3       1.00      1.00      1.00        33
           4       1.00      1.00      1.00        37
           5       1.00      1.00      1.00        25
           6       0.97      1.00      0.99        34
           7       1.00      1.00      1.00        24
           8       0.74      1.00      0.85        23
           9       0.96      1.00      0.98        23
          10       0.91      1.00      0.96        32
          11       1.00      0.96      0.98        26
          12       1.00      1.00      1.00        38
          13       1.00      0.86      0.93        22
          14       1.00      1.00      1.00        26
          15       1.00      1.00      1.00        30
          16       1.00      1.00      1.00        34
          17       1.00      0.97      0.99        37
          18       1.00      0.94      0.97        36
          19       1.00      1.00      1.00        34
          20       1.00      0.73      0.84        26
          21       1.00      1.00      1.00        28

    accuracy                           0.98       660
   macro avg       0.98      0.98      0.98       660
weighted avg       0.98      0.98      0.98       660
```




```
SVM Cross-Validation Accuracy: 97.34%
```

2) k-Nearest Neighbors (k-NN):

a) Accuracy: k-NN yielded a level of accuracy of 90.15%, marking it as the least effective within the models tested.

b) Cross-Validation Accuracy: The model recorded a cross-validation accuracy of 89.80%, indicating lower generalization capability.

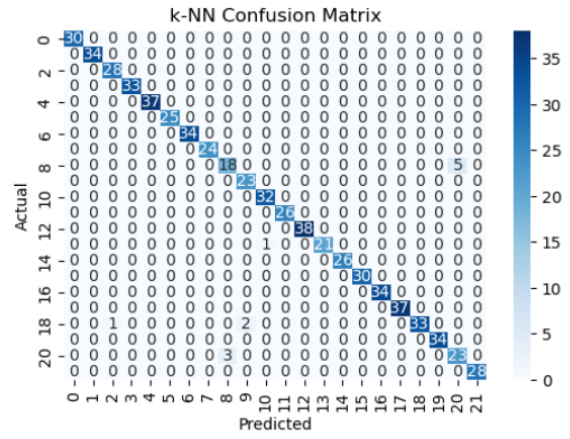
```
--- Evaluating k-NN ---

k-NN Accuracy: 98.18%

Classification Report for k-NN:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        30
           1       1.00      1.00      1.00        34
           2       0.97      1.00      0.98        28
           3       1.00      1.00      1.00        33
           4       1.00      1.00      1.00        37
           5       1.00      1.00      1.00        25
           6       1.00      1.00      1.00        34
           7       1.00      1.00      1.00        24
           8       0.86      0.78      0.82        23
           9       0.92      1.00      0.96        23
          10       0.97      1.00      0.98        32
          11       1.00      1.00      1.00        26
          12       1.00      1.00      1.00        38
          13       1.00      0.95      0.98        22
          14       1.00      1.00      1.00        26
          15       1.00      1.00      1.00        30
          16       1.00      1.00      1.00        34
          17       1.00      1.00      1.00        37
          18       1.00      0.92      0.96        36
          19       1.00      1.00      1.00        34
          20       0.82      0.88      0.85        26
          21       1.00      1.00      1.00        28

    accuracy                           0.98       660
   macro avg       0.98      0.98      0.98       660
weighted avg       0.98      0.98      0.98       660
```




```
k-NN Cross-Validation Accuracy: 97.34%
```

V. VISUAL FINDINGS

Correlation Heat Map: The correlation matrix identified a significant association between precipitation, temperature, and plant events that support its importance in the prediction task. The optimal model, gradient increase, SVM, and K-NN follow. Soil conditions. The results showed:

The Random Forest Classifier is the most efficient model for this, showing excellent accuracy of 98.95%, and robust performance through cross-validation and functional analysis. The K-NN showed moderate accuracy. This indicates limited effectiveness in the treatment of complex agricultural data records. The findings have had a significant impact on farmers and political decision makers, allowing data to make controlled decisions to improve crop yield and sustainability.

A. Future Research Could Be Investigated:

Test advanced algorithms such as the xgboost model and continue to activate accuracy and scalability. Verification of model applicability and improved generalizations.

REFERENCES

[1] Slingo, J. M., Challinor, A. J., Hoskins, B. J., & Wheeler, T. R. (2005). Food Crops in a Changing Climate. Philosophical Transactions: Biological Sciences, 360(1463), 1983–1989.
[2] Bandara, P., Weerasooriya, T., Nanayakkara, W. J. M., et al. (2020). Crop Recommendation System. International Journal of Computer Applications, 175(22).
[3] Jaslam, P. K. M., Kumar, M., Anurag, & Manjeet. (2020). EBLUP Estimate of Crop Yield at Sub-District Level in Hisar District, Haryana, India Using MODIS/Terra Data. Current Science, 119(12), 1982–1989.
[4] Zhai, T., Mohtar, R., El-Awar, F. A., Jabre, W., et al. (2004). Parameter Estimation for Process-Oriented Crop Growth Models. Transactions of the ASAE, 47(6), 2109–2119.
[5] Kumar, A., & Singh, P. (2021). Influence of Regional and Climatic Factors on Crop Growth and Quality.

